VFMM3D: Releasing the Potential of Image by Vision Foundation Model for Monocular 3D Object Detection

Bonan Ding, Jin Xie, Jing Nie, Jiale Cao, Xuelong Li, and Yanwei Pang

***Abstract*—Due to its cost-effectiveness and widespread availability, monocular 3D object detection, which relies solely on a single camera during inference, holds significant importance across various applications, including autonomous driving and robotics. Nevertheless, directly predicting the coordinates of objects in 3D space from monocular images poses challenges. Therefore, an effective solution involves transforming monocular images into LiDAR-like representations and employing a LiDAR-based 3D object detector to predict the 3D coordinates of objects. The key step in this method is accurately converting the monocular image into a reliable point cloud form. In this paper, we present VFMM3D, an innovative framework that leverages the capabilities of Vision Foundation Models (VFMs) to accurately transform single-view images into LiDAR point cloud representations. VFMM3D utilizes the Segment Anything Model (SAM) and Depth Anything Model (DAM) to generate high-quality pseudo-LiDAR data enriched with rich foreground information. Specifically, the Depth Anything Model (DAM) is employed to generate dense depth maps. Subsequently, the Segment Anything Model (SAM) is utilized to differentiate foreground and background regions by predicting instance masks. These predicted instance masks and depth maps are then combined and projected into 3D space to generate pseudo-LiDAR points. Finally, any object detectors based on point clouds can be utilized to predict the 3D coordinates of objects. Comprehensive experiments are conducted on two challenging 3D object detection datasets, KITTI and Waymo. Our VFMM3D establishes a new state-of-the-art performance on both datasets. Additionally, experimental results demonstrate the generality of VFMM3D, showcasing its seamless integration into various LiDAR-based 3D object detectors.**

***Index Terms*—3D object detection, vision foundation model, monocular vision.**

I. Introduction

The rise of autonomous systems has brought about an era where perceiving and understanding the environment in three dimensions is not just advantageous but imperative. Among the array of sensing methods, monocular vision emerges as a compelling choice due to its simplicity and cost-effectiveness. However, the challenge persists in extracting 3D information from a single 2D image, posing a significant obstacle. Monocular 3D object detection [1]–[5], tasked with estimating the 3D bounding boxes of objects from monocular images, stands as a pivotal element in numerous applications, spanning advanced driver-assistance systems (ADAS), robotics, and virtual reality, among others.

Traditional monocular 3D object detection approaches rely on geometry-based methods [6]–[11] or depth estimation techniques [12]–[18], which often suffer from a lack of accuracy and robustness, particularly in complex and dynamic scenes. The recent surge in deep learning has introduced new possibilities, with convolutional neural networks (CNNs) and vision transformers (ViTs) [19], revolutionizing the computer vision field. These models, pre-trained on extensive datasets, have demonstrated an uncanny ability to generalize and perform well across a multitude of tasks.

Vision Foundation Models (VFMs), such as the Depth Anything Model (DAM) [20] and the Segment Anything Model (SAM) [21], have shown particular promise in the domain of monocular 3D object detection. SAM with its ability to generate precise segmentation masks, and DAM with its capability to estimate precise scene depth from monocular images, provide a rich set of depth features that can significantly enhance the performance of monocular 3D detection models. The combination of these VFMs offers a unique opportunity to address the inherent challenges in monocular 3D detection, such as the ambiguity of depth and the need for robust segmentation.

In this paper, we present VFMM3D, a novel framework that synergistically integrates SAM and DAM to precisely perform monocular 3D object detection. Our approach leverages the strengths of both models to generate pseudo-LiDAR data enriched with semantic information and accurate depth. The ability of our framework may be improved further by more powerful vision foundation models and LiDAR-based 3D object detectors in the future. Based on the generated high-quality pseudo-LiDAR data, state-of-the-art LiDAR-based 3D object detectors could be employed to perform 3D object detection.

VFMM3D introduces several innovative components to the monocular 3D detection framework. Firstly, we propose a method for generating high-quality and foreground information-riched pseudo-LiDAR data that does not rely on being fine-tuned on specific datasets, making it widely applicable. Secondly, we incorporate a sparsification technique to address the computational inefficiency and noise associated with dense pseudo-LiDAR points. Lastly, our method adapts to various LiDAR-based 3D object detectors, showcasing its versatility and flexibility.

Through extensive experiments on the KITTI dataset, we demonstrate that VFMM3D surpasses existing state-of-the-art monocular 3D object detection methods across different difficulty levels. The superior performance of VFMM3D is attributed to its ability to effectively extract and utilize detailed 3D spatial information from monocular images by VFMs. Our work not only pushes the frontier of monocular 3D object detection but also provides a robust and generalizable solution that can be readily applied to real-world applications.

In conclusion, the contributions are as follows:

- To our knowledge, VFMM3D is the first approach that integrates vision foundation models with the monocular 3D object detection task. VFMM3D utilizes SAM and DAM without the need for being fine-tuned on specific datasets, in combination with any LiDAR-based 3D detector for monocular 3D object detection in arbitrary scenes.
- The Pseudo-LiDAR painting operation introduced in our methods enables better integration of results from SAM and DAM in 3D space, fully leveraging the 3D information that 2D images can provide for 3D tasks, thereby significantly improving the final detection accuracy.
- VFMM3D introduces a sparsification operation that enables seamless integration between the virtual point generation of visual foundation models and arbitrary 3D object detectors. It significantly enhances detection accuracy and substantially reduces computational costs and inference time.
- Extensive experiments on **KITTI** and **Waymo** datasets show our method achieves state-of-art results among existing monocular 3d object detection methods as shown in Fig. 2.

II. Related Work

Vision Foundation Models. Vision Foundation Models (VFMs) represent a significant advancement in the realm of computer vision, offering versatile solutions across various tasks owing to their robust pre-training on extensive datasets. Among VFMs, Vision Transformers (ViTs) [19] stand as pivotal models, trained on colossal datasets such as LVD-142M [22]. The efficacy of ViT is further amplified through approaches like DINO [23] and DINOv2 [22], which leverage self-supervised learning coupled with knowledge distillation techniques. One notable application of VFMs is evident in the Segment Anything Model (SAM) [21], which is particularly designed to be adept at generating precise masks for individual elements within images, showcasing its ability to handle detailed object segmentation. It's training on a vast dataset SA-1B encompassing 11 million images and 1.1 billion masks. Similarly, the Depth Anything Model (DAM) [20] emerges as a robust solution for monocular depth estimation, enabling accurate projection of 2D imagery into 3D space. This capability is instrumental in providing depth information for each pixel in an image, facilitating a more comprehensive understanding of scene geometry. The integration of SAM and DAM within the VFMM3D model represents a novel approach to enhancing monocular 3D object detection. SAM's segmentation abilities are harnessed to refine foreground information within pseudo-LiDAR data, while DAM's depth estimation capabilities aid in projecting 2D images into 3D space, thereby extracting as much valuable information for 3D spatial representation as possible from 2D images.

Monocular 3D Object Detection. In the field of monocular 3D object detection, a variety of methods have been developed to extract 3D information from a single 2D image. Deep3DBox [8] introduces the MultiBin approach for yaw estimation and utilizes geometric constraints of 2D bounding boxes to generate 3D bounding boxes. MonoPair [6] is a typical geometry-based method that uses spatial relationships between objects to estimate 3D properties. MonoFlex [24] proposes a flexible framework that separates truncated objects and amalgamates multiple depth estimation approaches, including direct regression and geometric solutions from keypoints. MonoDLE [16] and PGD [25] apply deep learning to directly predict 3D bounding boxes or depth on instances from images. MonoRUn [26] leverages self-supervised learning to establish dense correspondences and geometric relationships by using 3D bounding box annotations directly. MonoDTR [27] and MonoDETR [28] adaptive feature aggregation via a depth-guided transformer for monocular 3D object detection. Pseudo-LiDAR-based methods [29]–[32] generate a pseudo-LiDAR point cloud from the images and then process it using voxel-based techniques. CaDDN [17] constructs bird's-eye-view (BEV) representations by learning categorical depth distributions for each pixel. It then recovers bounding boxes from the BEV projection. MonoPSR [7] estimates instance point clouds and refines proposals by enforcing alignment between object appearance and the projected point cloud. QNet [33] transforms pseudo-LiDAR data into an image representation and employs powerful 2D CNNs to improve detection performance. To tackle this issue, M3D-RPN [12] introduces depth-aware convolutional layers within a 3D region proposal network to enhance the extracted features. RTM3D [34] estimates the projected vertices of 3D bounding boxes and resolves 3D properties through nonlinear least squares optimization. In this paper, we adopt a pseudo-LiDAR-based framework, considering its superior generalizability, which enables seamless integration with various detectors to accommodate diverse scenarios.

LiDAR-based 3D Object Detection. Within the domain of LiDAR-based 3D object detection, there are two primary paradigms: point-based and voxel-based models. PointR-CNN [35] exemplifies the former, harnessing PointNet++ [36], the pioneer of point-based methods, for feature extraction from raw LiDAR point clouds, followed by [37]–[39]. Contrarily, VoxelNet [40] showcases a robust voxel feature encoding layer, transforming 3D LiDAR point clouds into equidistantly spaced 3D voxels. Building upon this voxel-centric approach, SECOND [41] introduces a novel sparse 3D convolution layer tailored for LiDAR point clouds. Departing from traditional voxel methods, PointPillars [42] employs pillars for feature encoding, enhancing network efficiency. Similarly, Voxel-RCNN [43] proposes voxel query and voxel RoI pooling methods to fully exploit voxel features in a two-stage framework. Embracing a hybrid strategy, PV-RCNN [44] integrates point and voxel advantages, employing voxel set abstraction to fuse multi-scale voxel features into sampled key points.

III. Method

A. Task Definition

The goal of monocular 3D object detection is to classify and localize the objects of interest in 3D space by giving a single

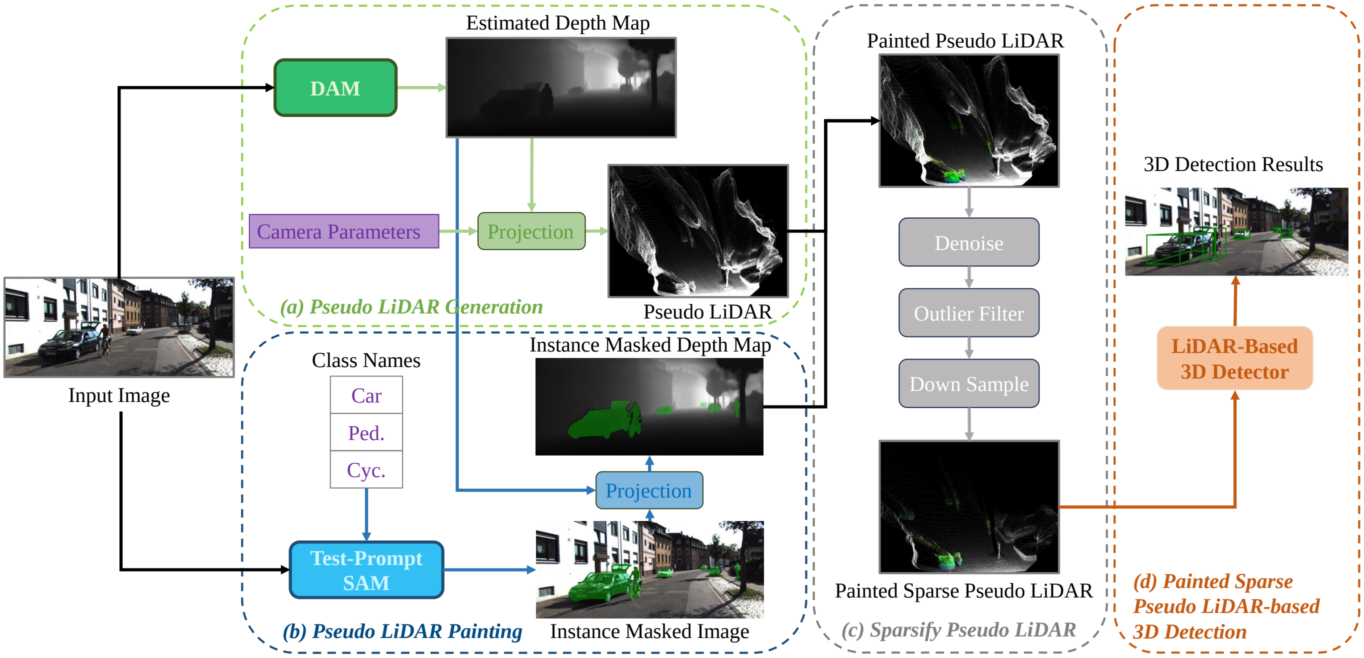

Fig. 1. The overall architecture of VFMM3D. Our model consists of four parts: (a) Pseudo-LiDAR generation by DAM, (b) Pseudo-LiDAR painting by text-prompt SAM, (c) Pseudo-LiDAR sparsification, (d) 3D object detection by LiDAR-based detectors.

view RGB image and its corresponding camera parameters. Each object is represented by its category score and bounding box (BBox) which is parameterized by its center coordinates (x, y, z), size (h, w, l), and orientation θ.

B. Framework Overview

Existing monocular 3D object detection methods based on pseudo-LiDAR often require pre-trained depth estimation and semantic segmentation models pre-trained on specific datasets. To generate high-quality pseudo-LiDAR data without the need for pre-training on specific datasets and applicable across any datasets, we introduce VFMM3D leveraging two Visual Foundation Models, DAM and SAM. DAM is tasked with generating robust pseudo-LiDAR across arbitrary datasets and scenes. However, in comparison to real LiDAR data, the pseudo-LiDAR produced by this depth prediction still lacks precision. To address this issue, we integrate SAM to furnish accurate foreground information to the pseudo-LiDAR, alleviating the inaccuracies stemming from imprecise depth estimation. Moreover, conventional LiDAR-based 3D object detectors are designed for sparse point cloud data. To adapt to existing detectors while bolstering detection speed and mitigating potential accuracy impacts from redundant and inaccurate pseudo-LiDAR points, we propose a sparsification method aimed at enhancing efficiency and accuracy. The whole pipeline of VFMM3D is shown in Fig. 1 and Algorithm 1, which mainly contains four steps:

- **Pseudo-LiDAR Generation**. At this step, we begin by acquiring the depth map using the DAM. Next, we project this depth map into 3D space and obtain pseudo-LiDAR data.
- **Pseudo-LiDAR Painting**. SAM takes the RGB image as the input and combines text prompts to perform foreground object segmentation. The foreground object segmentation results are mapped to the depth map obtained by the DAM, which is employed to highlight the foreground object depth map and filter noises. Then the corresponding pseudo-LiDAR is also more accurate.
- **Pseudo-LiDAR Sparsification**. Since the number of pseudo-LiDAR points obtained from the depth map is much larger than the LiDAR point cloud in the real scene, we adopt a sparsification step to appropriately sparse the painted pseudo-LiDAR to adapt to the final LiDAR-based 3d object detector.

Algorithm 1 VFMM3D

```
Input:
  RGB image $I \in \mathbb{R}^{H \times W \times 3}$.

Output:
  Predicted 3D bounding boxes $B_{3D} = \{(x_i, y_i, z_i, h_i, w_i, l_i)\}_{i=1}^{M}$

  $D = Depth\ Generation(I)$   ▷ Depth Map $D \in \mathbb{R}^{H \times W \times 1}$
  $P = Projection(D)$   ▷ Pseudo-LiDAR $P \in \mathbb{R}^{N \times 3}$
  $I_M = Mask\ Generation(I, B)$   ▷ Instance Mask $I_M \in bool^{H \times W}$
  $\hat{P} = Painting(D, I_M, P)$   ▷ Painted Pseudo-LiDAR $\hat{P} \in \mathbb{R}^{N \times 6}$
  $\bar{P} = Sparsification(P)$
        ▷ Sparse Painted Pseudo-LiDAR $\bar{P} \in \mathbb{R}^{\bar{N} \times 6}$
  $B_{3D} = LiDAR\text{-}Based\ 3D\ Detector(\bar{P})$
```

- **LiDAR-based 3D Detection.** To perform 3D object detection using the painted pseudo-LiDAR, we employ a LiDAR-based 3D object detector. This detector takes the painted pseudo-LiDAR generated from the previous steps as input and produces the final detection results.

C. Pseudo-LiDAR Generation with DAM

Preliminaries. The foundational step in our monocular 3D object detection pipeline is the estimation of depth from single-view images. We introduce the **Depth Anything Model (DAM)** [20] for robust monocular depth estimation (MDE). DAM does not rely on technical modules but instead focuses on scaling up the dataset with a novel data engine designed to automatically collect and annotate approximately 62 million unlabeled images from various public large-scale datasets. The extensive use of unlabeled data significantly broadens the data coverage, thereby reducing the generalization error and improving the model's ability to handle diverse and challenging scenes.

The DAM leverages a two-pronged strategy to enhance its performance. First, it creates a more challenging optimization target through the use of data augmentation tools, compelling the model to actively seek additional visual knowledge and acquire robust representations. Second, it incorporates an auxiliary supervision mechanism that enforces the model to inherit rich semantic priors from pre-trained encoders. The above-mentioned approach not only enhances the model's performance in depth estimation but also provides a multi-task encoder capable of handling both middle-level and high-level perception tasks.

Depth Estimation by DAM. DAM is further fine-tuned with metric depth information from standard datasets such as NYUv2 and KITTI, setting new state-of-the-art records in metric depth estimation accuracy. Given a single image $I \in \mathbb{R}^{H\times W\times 3}$, we can get a reliable depth map $D \in \mathbb{R}^{H\times W\times 1}$ generated by the DAM. After getting the depth map, we then project it to 3d space to get the pseudo-LiDAR in real-world coordinate $P = \{(x^{(n)}, y^{(n)}, z^{(n)})\}_{n=1}^{N}$ by the first project the depth map to the camera coordinate system:

$$\begin{cases} z_c = d, \\ x_c = \frac{(u-C_x)\times z}{f}, \\ y_c = \frac{(v-C_y)\times z}{f}, \end{cases} \tag{1}$$

where d is the estimated depth of pixel (u, v) in the depth map, and (C_x, C_y) is the principal point of camera. f_x and f_y is the camera focal length along x and y axes, respectively. N is the number of pixels. Then given the camera extrinsic matrix M_E:

$$M_E = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}, \tag{2}$$

where M_E is a 4×4 matrix. R is a 3×3 orthogonal unit matrix, also known as a rotation matrix. t is a three-dimensional translation vector. Then we can get pseudo-LiDAR in the world coordinate system by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M_E^{-1} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{3}$$

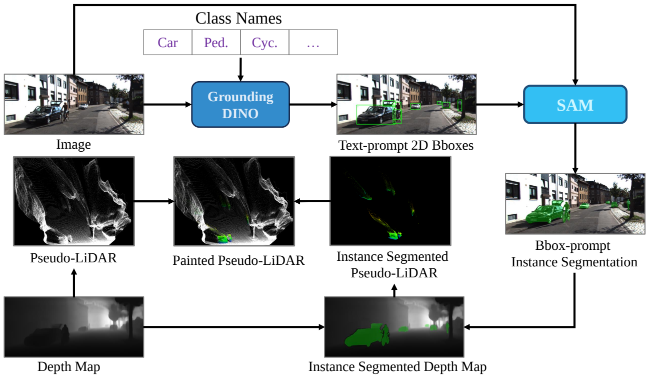

Fig. 2. The detailed architecture of Pseudo-LiDAR Painting Text-Prompt SAM.

D. Pseudo-LiDAR Painting with Text-Prompt SAM

Preliminaries. The **Segment Anything Model (SAM)** [21] represents a significant leap in the field of computer vision as a VFM, trained on an extensive SA-1B dataset, which comprises over 11 million images and an astounding 1 billion masks, endowing it with remarkable generalization capabilities across different scenes and object types. The design of SAM is underpinned by a lightweight mask decoder and a powerful image encoder, which work in tandem to process prompts and produce high-quality segmentation outputs. Its promptable nature allows for flexibility in accepting various forms of input, including points, bounding boxes, and even free-form text, making it accessible for a multitude of applications. The SAM family methods [45], [46] further expand on this foundation, exploring different configurations and adaptations that enhance the core capabilities of SAM. These methods have been evaluated extensively, showcasing impressive zero-shot performance and the ability to compete with, or even surpass, fully supervised models in various segmentation tasks. SAM and its family methods cannot be directly applied to 3D scenes due to the inherent gap between 2D and 3D spatial representations. However, applying them to monocular 3D object detection can naturally provide robust and powerful semantic-level image features for this task.

Depth Map Segmentation by Text-prompt SAM. As described in PointPainting [47], MonoPseudo [31] and Fusion-painting [48], the semantic information extracted from the foreground objects in an image through semantic segmentation plays a crucial role in enhancing the accuracy and robustness of 3D object detection. To showcase the broad adaptability of our method, we avoid the use of 2D object detection models that have been pre-trained specifically on autonomous driving data sets, as well as any additional prior data. As a result, it is unable to offer SAM the usual types of cues such as points, bounding boxes, or masks. Instead, the SAM used in this approach is based on text-based prompts. Specifically, the text-prompt SAM contains two parts: a text-prompt open-set 2d object detector **Grounding DINO** [49]

Algorithm 2 Pseudo-LiDAR Painting with Text-Prompt SAM

```
Input:
  RGB image $I \in \mathbb{R}^{H\times W\times 3}$.
  Depth map $D \in \mathbb{R}^{H\times W\times 1}$.
  Pseudo-LiDAR points $P \in \mathbb{R}^{N\times 3}$ projected from $D$.

Output:
  Painted pseudo-LiDAR points $\hat{P} \in \mathbb{R}^{N\times 3+3}$

  $B_{2D} = GroundingDINO(I)$                ▷ $B_{2D} \in \mathbb{R}^{K\times 4}$
  $I_M = SAM(I, B_{2D})$                     ▷ $I_M \in bool^{H\times W}$
  $D_M = D \times I_M$

  for $d \in D_M, p \in P$ do
      if $d \neq 0$ then
          $i = I[u, v, :]$                   ▷ $i \in \mathbb{R}^3$
          $\hat{p}$ =Concatenates$(p, i)$    ▷ $\hat{p} \in \mathbb{R}^6$
      else
          $\hat{p}$ =Concatenates$(p, \mathbf{0}_3)$    ▷ $\hat{p} \in \mathbb{R}^6$
      end if
  end for
```

and a **SAM** as shown in Fig. 2. First, the Grounding DINO takes RGB image I in and get 2d object detection BBoxes $B_{2D} = \{(u_B^{(k)}, v_B^{(k)}, h_B^{(k)}, l_B^{(k)})\}_{k=1}^{K}$ by text-prompt (*i.e.*Car, Pedestrian and Cyclist on KITTI dataset.), where K is the number of predicted BBoxes and u_B, v_B, h_B, l_B denote the center pixel coordinates and the size of each predicted BBox, respectively. Then put the RGB image I and predicted 2d BBoxes B_{2D} into SAM, and set the BBoxes B_{2D} as prompt to get the semantic segmentation masked image I_M. Then we project the mask of I_M onto its corresponding depth map D generated from DAM to get the foreground masked depth map D_M. As inspired by Pointpainting [47], we add the RGB-channel of pixels in D_M to its corresponding pseudo-LiDAR P to get the painted pseudo-LiDAR $\hat{P}$. The details of the painting algorithm are shown in Algorithm 2.

E. Sparsify Pseudo-LiDAR

Many recent monocular [29], [31], [32] or multimodal fusion-based [50]–[52] 3D object detectors utilize pseudo-LiDAR generated from images through depth estimation or completion algorithms. However, pseudo-LiDAR is much denser compared to true LiDAR captured by scans, as it has an equal number of pseudo-points as the number of pixels in the image. Additionally, as mentioned in VirConv [52], the original pseudo-LiDAR introduces a significant computational cost, leading to a decrease in detection speed (more than 2 3× slower than true LiDAR-based 3D object detectors). Furthermore, too many background points may distract the networks from the foreground objects since only the points corresponding to the foreground objects are the focus in LiDAR-based 3D object detectors. Besides the issues mentioned above, pseudo-LiDAR also suffers from noise problems. These noises stem from inaccuracies in depth estimation, resulting in non-Gaussian distributed noise. Traditional methods struggle to eliminate such noise. In Pseudo-LiDAR++ [50], cheap 2-beams true LiDAR point clouds are used for pseudo-LiDAR correction to mitigate these noises inexpensively. However, our proposed method relies solely on single-image data. Therefore, we employ some traditional methods to minimize the impact of these noises as much as possible. The sparsify operation is shown in Fig. 1. Specifically, after getting the painted pseudo-LiDAR $\hat{P}$, we first map each point in $\hat{P}$ to spherical coordinate space and voxelize the $\hat{P}$ with a small voxel size using the coordinates in this spherical space. We retain only the mean 3d coordinates of all points within each voxel to get the denoised painted pseudo-LiDAR $\hat{P}_d$. Then we filter out outliers pseudo points in $\hat{P}_d$ based on predefined ranges. The specific filtering ranges will be set differently according to different datasets, which will be detailed in Section IV. Finally, we voxelize based on the 3D coordinates in the pseudo-LiDAR using a larger voxel size, and then randomly sample points within each voxel to be less than or equal to a fixed number of points (5 in this method), we can get the final sparse painted pseudo-LiDAR $\bar{P} \in \mathbb{R}^{\bar{N}\times 6}, \bar{N} \ll N$.

F. LiDAR-Based Detection

The painted pseudo-LiDAR $\bar{P}$ can be input into any LiDAR-based 3D object detector to obtain 3D object results. In this work, we focus on LiDAR-only 3D detectors, although our method allows for the use of any multimodal fusion (i.e., inputting LiDAR + image) based 3D detector. This decision is made for the sake of detection speed, considering that the pseudo-LiDAR point count is significantly higher compared to real point counts. Employing a multimodal fusion-based detector would substantially decrease detection speed. Therefore, we only consider single LiDAR-based 3D detectors in this work.

Specifically, we show that VFMM3D is compatible with three different LiDAR-based detectors: PV-RCNN [44], Voxel-RCNN [43] and PointPillars [42]. These include widely-used LiDAR detectors, each featuring a distinct network architecture: single-stage (PointPillars) versus two-stage (PV-RCNN and Voxel-RCNN).

Voxel-RCNN. The Voxel-RCNN method is designed to enhance the performance of voxel-based 3D object detection while maintaining computational efficiency. It introduces a two-stage framework that includes a 3D backbone network for feature extraction, a 2D bird-eye-view (BEV) Region Proposal Network (RPN), and a detection head. The key innovation is voxel RoI pooling, which straightforwardly extracts RoI features from voxel features for subsequent refinement. This approach allows for real-time frame processing while achieving comparable detection accuracy to that of state-of-the-art point-based approaches, significantly reducing computation costs.

PV-RCNN. PV-RCNN (Point-Voxel Feature Set Abstraction for 3D Object Detection) is a high-performance 3D object detection framework that deeply integrates 3D voxel Convolutional Neural Networks with PointNet-based set abstraction. It leverages the efficient learning of the 3D voxel CNN for high-quality proposals and the flexible receptive fields of PointNet-based networks for capturing accurate location information

TABLE I

Comparsion Result on Waymo *val.* set for the vehicles class. Red indicates the best results, while blue indicates the second-best results.

Diff.	Extra	Method	Reference	3D mAP / mAPH(IoU=0.7)				3D mAP / mAPH(IoU=0.5)			
				Overall	0 - 30m	30 - 50m	50m - ∞	Overall	0 - 30m	30 - 50m	50m - ∞
L1	LiDAR	CaDDN [17]	CVPR 21	5.03 / 4.99	14.54 / 14.43	1.47 / 1.45	0.10 / 0.10	17.54 / 17.31	45.00 / 44.46	9.24 / 9.11	0.64 / 0.62
	LiDAR	MonoNeRD [53]	ICCV 23	10.66 / 10.56	27.84 / 27.57	5.40 / 5.36	0.72 / 0.71	31.18 / 30.70	61.11 / 60.28	26.08 / 25.71	6.60 / 6.47
	LiDAR	DID-M3D [54]	ECCV 22	- / -	- / -	- / -	- / -	20.66 / 20.47	40.92 / 40.60	15.63 / 15.48	5.35 / 5.24
	Depth	PatchNet [55]	ECCV 20	0.39 / 0.37	1.67 / 1.63	0.13 / 0.12	0.03 / 0.03	2.92 / 2.74	10.03 / 9.75	1.09 / 0.96	0.23 / 0.18
	Depth	PCT [56]	NeurIPS 21	0.89 / 0.88	3.18 / 3.15	0.27 / 0.27	0.07 / 0.07	4.20 / 4.15	14.70 / 14.54	1.78 / 1.75	0.39 / 0.39
	-	M3D-RPN [12]	ICCV 19	0.35 / 0.34	1.12 / 1.10	0.18 / 0.18	0.02 / 0.02	3.79 / 3.63	11.14 / 10.70	2.16 / 2.09	0.26 / 0.21
	-	GUPNet [9]	ICCV 21	2.28 / 2.27	6.15 / 6.11	0.81 / 0.80	0.03 / 0.03	10.02 / 9.94	24.78 / 24.59	4.84 / 4.78	0.22 / 0.22
	-	DEVIANT [57]	ECCV 22	2.69 / 2.67	6.95 / 6.90	0.99 / 0.98	0.02 / 0.02	10.98 / 10.89	26.85 / 26.64	5.13 / 5.08	0.18 / 0.18
	-	MonoJSG [58]	CVPR 22	0.97 / 0.95	4.65 / 4.59	0.55 / 0.53	0.10 / 0.09	5.65 / 5.47	20.86 / 20.26	3.91 / 3.79	0.97 / 0.92
	-	SSD-MonoDETR [59]	TIV 2023	4.54 / -	9.93 / -	1.18 / -	0.15 / -	11.83 / -	27.69 / -	5.33 / -	0.85 / -
	-	VFMM3D(Voxel-RCNN)	-	7.48 / 7.24	17.41 / 16.91	2.08 / 1.99	0.28 / 0.24	18.26 / 17.46	38.26 / 36.75	8.35 / 7.95	1.60 / 1.36
	-	VFMM3D(PV-RCNN)	-	8.06 / 7.79	19.38 / 18.81	2.22 / 2.05	0.15 / 0.12	18.39 / 17.48	39.26 / 37.66	8.81 / 8.06	1.10 / 0.87
L2	LiDAR	CaDDN	CVPR 21	4.49 / 4.45	14.50 / 14.38	1.42 / 1.41	0.09 / 0.09	16.51 / 16.28	44.87 / 44.33	8.99 / 8.86	0.58 / 0.55
	LiDAR	MonoNeRD [53]	ICCV 23	10.03 / 9.93	27.75 / 27.48	5.25 / 5.21	0.60 / 0.59	29.29 / 28.84	60.91 / 60.08	25.36 / 25.00	5.77 / 5.66
	LiDAR	DID-M3D [54]	ECCV 22	- / -	- / -	- / -	- / -	19.37 / 19.19	40.77 / 40.46	15.18 / 15.04	4.69 / 4.59
	Depth	PatchNet [55]	ECCV 20	0.38 / 0.36	1.67 / 1.63	0.13 / 0.11	0.03 / 0.03	2.42 / 2.28	10.01 / 9.73	1.07 / 0.94	0.22 / 0.16
	Depth	PCT [56]	NeurIPS 21	0.66 / 0.66	3.18 / 3.15	0.27 / 0.26	0.07 / 0.07	4.03 / 3.99	14.67 / 14.51	1.74 / 1.71	0.36 / 0.35
	-	M3D-RPN [12]	ICCV 19	0.33 / 0.33	1.12 / 1.10	0.18 / 0.17	0.02 / 0.02	3.61 / 3.46	11.12 / 10.67	2.12 / 2.04	0.24 / 0.20
	-	GUPNet [9]	ICCV 21	2.14 / 2.12	6.13 / 6.08	0.78 / 0.77	0.02 / 0.02	9.39 / 9.31	24.69 / 24.50	4.67 / 4.62	0.19 / 0.19
	-	DEVIANT [57]	ECCV 22	2.52 / 2.50	6.93 / 6.87	0.95 / 0.94	0.02 / 0.02	10.29 / 10.20	26.75 / 26.54	4.95 / 4.90	0.16 / 0.16
	-	MonoJSG [58]	CVPR 22	0.91 / 0.89	4.64 / 4.65	0.55 / 0.53	0.09 / 0.09	5.34 / 5.17	20.79 / 20.19	3.79 / 3.67	0.85 / 0.82
	-	SSD-MonoDETR [59]	TIV 23	4.12 / -	8.87 / -	1.02 / -	0.13 / -	11.34 / -	27.62 / -	5.21 / -	0.76 / -
	-	VFMM3D(Voxel-RCNN)	-	6.62 / 6.41	17.17 / 16.67	1.89 / 1.81	0.22 / 0.19	16.19 / 15.48	37.77 / 36.29	7.58 / 7.21	1.25 / 1.06
	-	VFMM3D(PV-RCNN)	-	7.14 / 6.89	19.11 / 18.55	2.01 / 1.86	0.12 / 0.09	16.31 / 15.50	38.78 / 37.19	8.01 / 7.33	0.86 / 0.68

TABLE II

Comparsion of our model with state-of-the-art models on the KITTI *val.* set for the car class. 'Mod.' indicates the moderate difficulty level. * means the training set of model's depth estimator may overlap the validation set. Red indicates the best results, while blue indicates the second-best results.

Method	Reference	3D AP@0.7			BEV AP@0.7			3D AP@0.5			BEV AP@0.5		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Monopair [6]	CVPR 2020	16.28	12.30	10.42	24.12	18.17	15.76	55.38	42.39	37.99	61.06	47.63	41.92
MonoDLE [16]	CVPR 2021	17.45	13.66	11.68	24.97	19.33	17.01	55.41	43.42	37.81	60.73	46.87	41.89
MonoFlex [24]	CVPR 2021	24.22	17.34	15.13	31.65	23.29	20.02	60.70	45.65	39.91	66.26	49.30	44.42
GUPNet [9]	ICCV 2021	22.76	16.46	13.72	31.07	22.94	19.75	57.62	42.33	37.59	61.78	47.06	40.88
DDMP-3D* [60]	CVPR 2021	28.12	20.39	16.34	-	-	-	-	-	-	-	-	-
CaDNN [17]	CVPR 2021	23.57	16.31	13.84	-	-	-	-	-	-	-	-	-
MonoRUn [26]	CVPR 2021	20.02	14.65	12.61	-	-	-	59.71	43.39	38.44	-	-	-
HomoLoss [61]	CVPR 2022	23.04	16.89	14.90	31.04	22.99	19.84	-	-	-	-	-	-
MonoDDE [62]	CVPR 2022	26.66	19.75	16.72	35.51	26.48	23.07	-	-	-	-	-	-
MonoDTR [27]	CVPR 2022	24.52	18.57	15.51	33.33	25.35	21.68	64.03	47.32	42.20	69.04	52.47	45.90
MonoGround [18]	CVPR 2022	25.24	18.69	15.58	32.68	24.79	20.56	62.60	47.85	41.97	67.36	51.83	45.65
OPA-3D [63]	IRAL 2023	24.97	19.40	16.59	33.80	25.51	22.13	-	-	-	-	-	-
MonoNeRD [53]	ICCV 2023	20.64	15.44	13.99	29.03	22.03	19.41	-	-	-	-	-	-
PDR [4]	TCSVT 2023	27.65	19.44	16.24	35.59	25.72	21.35	-	-	-	-	-	-
VFMM3D(PointPillars)	-	22.43	15.45	13.92	35.85	24.88	22.75	63.52	45.60	41.79	68.26	48.62	44.75
VFMM3D(Voxel-RCNN)	-	29.09	19.41	17.09	39.43	26.56	23.69	68.72	50.53	45.02	73.17	52.90	47.25
VFMM3D(PV-RCNN)	-	29.05	19.10	16.86	41.78	28.53	25.61	69.95	51.78	47.01	74.24	54.98	49.69
VFMM3D(PointPillars)*	-	27.23	17.15	13.95	38.02	24.93	20.79	69.94	48.75	42.98	74.12	54.37	48.09
VFMM3D(Voxel-RCNN)*	-	34.60	21.58	18.23	44.18	28.66	24.02	72.85	51.83	45.53	76.69	55.43	48.95
VFMM3D(PV-RCNN)*	-	32.06	21.00	17.49	43.30	28.88	24.72	73.21	53.51	46.88	77.09	58.37	51.69

and context. The framework summarizes the 3D scene into a small set of keypoints via a novel voxel set abstraction module and then aggregates these keypoint features to RoI-grid points for proposal refinement. This integration of point-based and voxel-based feature learning leads to improved performance in 3D object detection with manageable memory consumption.

PointPillars. PointPillars is a method designed for efficient object detection from point clouds, particularly for autonomous driving applications. It introduces a novel encoder that employs PointNets to learn representations of point clouds organized into vertical columns or pillars. The encoded features can be integrated with any standard 2D convolutional detection architecture, and PointPillars introduces a streamlined downstream network for this integration. By using pillars rather than voxels, PointPillars eliminates the need for hand-tuning the binning of the vertical direction and leverages the efficiency of 2D convolutions on a GPU. The method significantly outperforms previous encoders in both speed and accuracy, running at 62 Hz and setting new standards for performance on KITTI benchmarks for both 3D and bird's eye view detection.

IV. Experiments

A. Settings

Datasets. We evaluate our method on two datasets, **Waymo** and **KITTI**. The Waymo dataset [64] contains 1,150 video sequences collected from diverse driving environments. We use the official splitting protocol to split the dataset into a train set with 798 sequences, 158081 samples and a validation set with

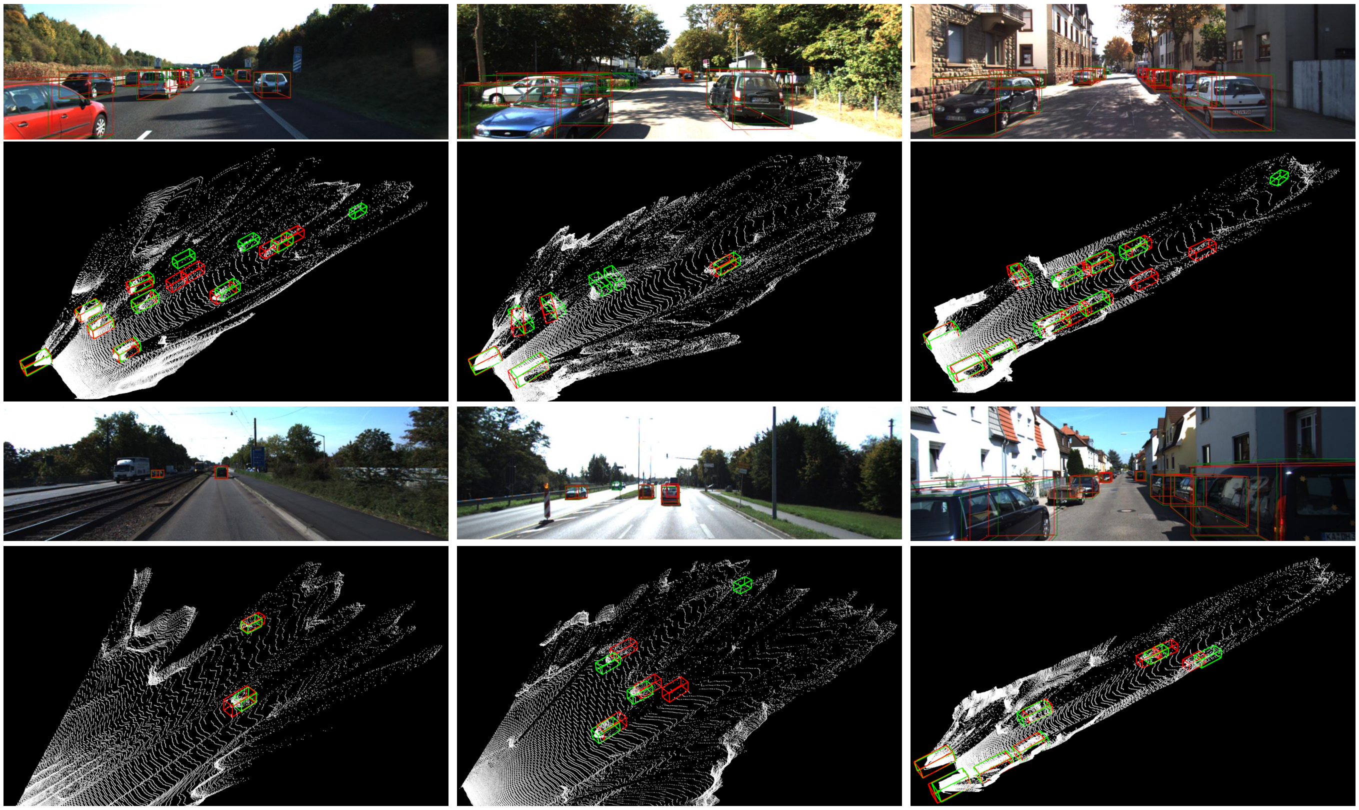

Fig. 3. Qualitative results of VFMM3D on KITTI *val.* set. We visualize our 3D bounding box estimates (in red) alongside ground truth annotations (in green) on front view images (1st and 3rd rows) and pseudo-LiDAR point clouds (2nd and 4th rows).

202 sequences, 39,848 samples. We follow DEVIANT [57], PCT [56] and other methods [53], [58], using the front view camera for monocular 3D object detection. The KITTI dataset contains two benchmarks, 3D object detection and bird's eye viewk [65], which comprises 7,481 training samples and 7,518 testing samples, along with the corresponding LiDAR point clouds, stereo images, and full camera matrix. We divided the original training samples into a *train* spilt set with 3,712 samples and a *validation* split set with 3,769 samples following the previous methods [29], [31]. It is worth noting that during both training and testing phases, our approach exclusively utilizes single-view RGB image data and does not incorporate any LiDAR point cloud or stereo image data.

Evaluation Metrics. On Waymo, we follow the official evaluation metrics, evaluate on two object levels: Level 1 and Level 2 with mAP and $mAPH$. It assigns each object to a level according to the number of LiDAR points contained within its 3D bounding box. Besides, we also provide the performance on three distance ranges: [0m, 30m), [30m, 50m), [50m, ∞]. For KITTI, we report the performance of our method with the standard evaluation metric AP_{40} [66], in terms of average precision sampled at 40 recall positions in the precision-recall curve, under three different difficulties (easy, moderate, and hard). For broad validation, we selected two IoU thresholds for car class on 3D boxes and BEV boxes, namely 0.7 (AP@0.7) and 0.5 (AP@0.5).

Training Details. The size of images is set to 1224×370 pixels on the KITTI dataset and 1920× 1280 pixels on the Waymo dataset without any 2D data augmentation. In the LiDAR-based detection stage, we use 3D random horizontal flip augmentation and point-shuffle augmentation. Due to the inaccuracy of depth estimation, there is a significant deviation between the pseudo points inside the 3D annotated boxes and the real points. Therefore, in this method, we do not utilize the data augmentation method of GT-sampling [41]. We use Voxel-RCNN [43] as the LiDAR-based 3D object detector for experiments on the KITTI dataset. For Voxel-RCNN and PV-RCNN [44], we set the pseudo-LiDAR range as [0, 70.4], [-40.0, 40.0], [-3.0, 1.0] meters along the X, Y, and Z axes, respectively. And the voxel size is (0.05, 0.05, 0.1) meters. For PointPillars [42], we set the pseudo-LiDAR range as [0, 69.12], [-39.68, 39.68], [-3.0, 1.0] meters, the pillar size is set as (0.16, 0.16, 4) meters. On the Waymo dataset, we use Voxel-RCNN and PV-RCNN and set the pseudo-LiDAR range as [0, 75.2], [-75.2, 75.2], [-2.0, 4.0] meters with voxel size (0.1, 0.1, 0.15) meters and [0, 59.6], [-25.6, 25.6], [-2.0, 4.0] meters with voxel size (0.05, 0.05, 0.15) meters, respectively.

Our method is primarily implemented using OpenPCDet [67]. We utilize pre-trained models, including Grounding DINO, SAM with vit-h, and DAM with vit-l, which is finetuned on KITTI outdoor metric depth datasets for containing metric depth estimation ability. However, we note the officially provided DAM model is finetuned on the whole KITTI-depth dataset, as mentioned in DD3D [30] and Pesuo-Lidar++ [50], this training set overlap with the KITTI-3D validation data for detection. Thus, we use the KITTI-3D clean training split,

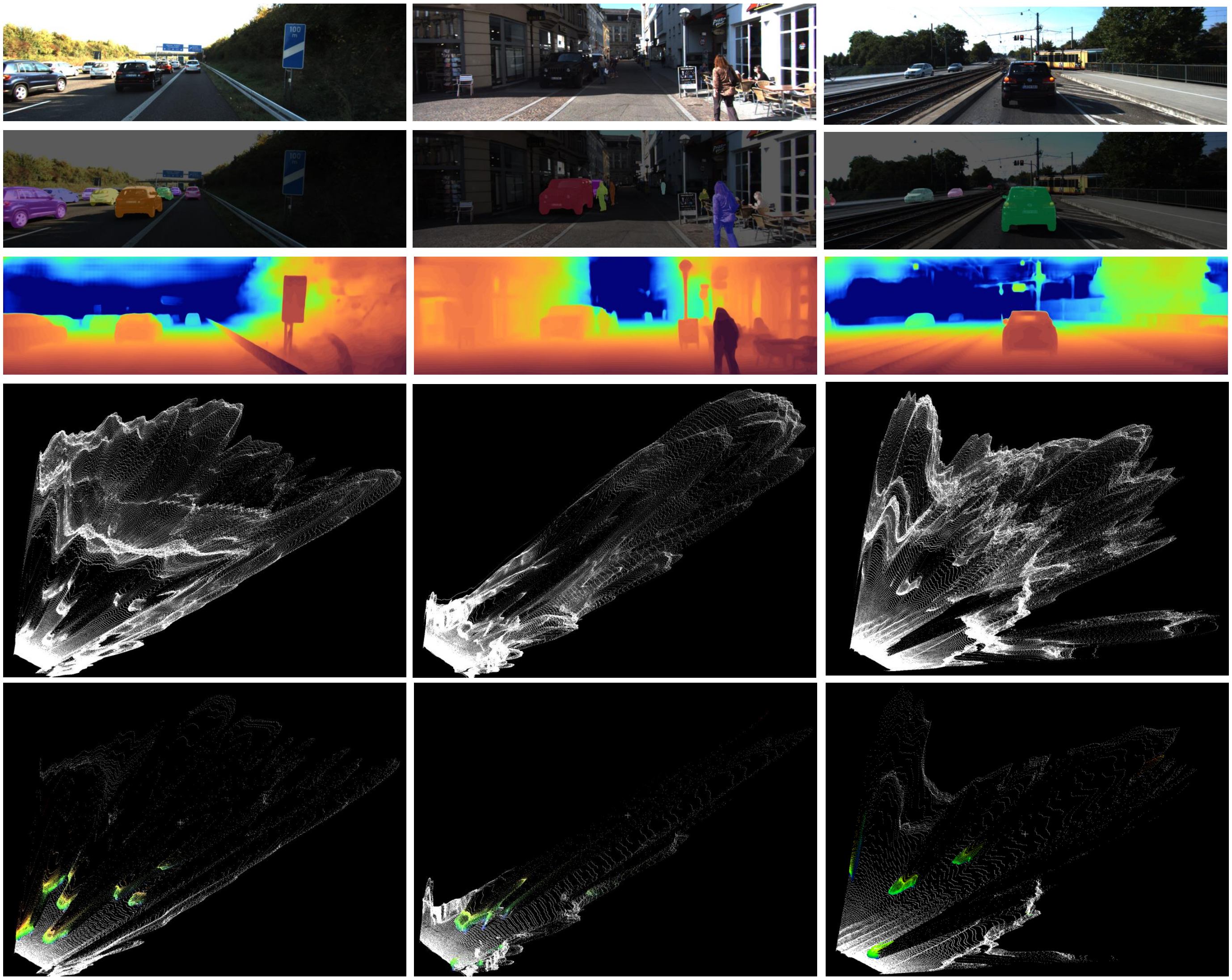

Fig. 4. Visualization of segmentation results (2nd row) from text-prompt SAM, depth maps (3rd row) from DAM, raw Pseudo-LiDAR (4th row) from depth map and Painted Sparse Pseudo-LiDAR (5th row) results from each component of VFMM3D on KITTI.

which removes training images that are geographically close to any of the KITTI-3D images provided by DD3D to finetune the DAM to avoid any bias. We also demonstrate the results on KITTI with DAM finetuned with the whole KITTI-depth dataset as some methods [60] also report this kind of results, as shown in Tbale II with the indicator *. The LiDAR-based 3D object detectors are trained on the sparse painted pseudo-LiDAR using 8 RTX A6000 GPUs. We employ the AdamW [68] optimizer with a learning rate of 4e-4 and weight decay of 0.01. The batch size is set to 32 for PointPillars, 32 for Voxel-RCNN, and 8 for PV-RCNN. Additionally, the number of training epochs is set to 80 for all detectors.

B. *Waymo Results*

Tab. I shows the performance comparison results on Waymo validation spilt. It demonstrates our method VFMM3D outperforms other monocular methods without extra modality data during training [9], [12], [57], [58], which surpasses the latest monocular method DEVIANT [57] by 7.41% and 6.20% on level 1 and level 2 3D mAP(IoU=0.5), respectively. It is worth noting that our method even outperforms CaDNN [17], which uses LiDAR as extra training data, 0.72% on level 1 3D mAP(IoU=0.5).

C. *KITTI Results*

We compare the performance of our method on the KITTI *val.* set for the car class with the state-of-the-art methods. The related results are shown in Tab. II. It is evident that our method outperforms existing methods in both 3D and BEV under all difficulty levels. Specifically, when IoU is set to 0.5, our method surpasses these methods by even greater margins. Compared to the latest monocular object detection method MonoNeRD [53], our approach surpasses it by 8.41% and 12.75% in 3D and BEV at the easy level, respectively. Some qualitative results of 3D detection boxes from our method on KITTI *val.* set are visualized in Fig. 3.

TABLE III

Ablation analysis on KITTI *val.* set. We quantify the impacts of painting strategy, *RGB P* means painting all pseudo-LiDAR with corresponding RGB pixel, *SAM P* refers to pseudo-LiDAR painting by text-prompt SAM.

DAM	RGB-P	SAM-P	3D AP@0.7			BEV AP@0.7		
			Easy	Mod.	Hard	Easy	Mod.	Hard
✓			21.50	15.88	14.05	34.81	25.38	22.79
✓	✓		26.96	17.77	15.77	36.63	25.60	23.51
✓		✓	29.05	19.10	16.86	41.78	28.53	25.61

D. Visualization of Each Component Results.

We visualize the results of each component in VFMM3D in Fig. 4. In the second and third rows, both Text-prompt SAM and DAM produce good segmentation results and accurate depth maps. Comparing the fourth and fifth rows, it is evident that our sparse pseudo-LiDAR obtained by our designed sparsify and painting operations can reduce a significant amount of useless noise points when enhancing information for foreground points, as opposed to the dense pseudo-LiDAR directly derived from the depth map.

E. Ablation Study

In this section, we delve into our method from two key perspectives: the impacts of the pseudo-LiDAR painting strategy and the ablation study of different LiDAR-based 3D object detectors.

Impacts of Different Painting Strategy. We conduct ablation experiments to assess the impacts of different pseudo-LiDAR painting strategies on the final 3D detection results with the PV-RCNN 3D detector. As depicted in Tab. III, we observe that the detailed and accurate foreground information provided by the text-prompt SAM (*SAM-P*) enables subsequent LiDAR-based 3D detectors to concentrate on the features of foreground objects efficiently. In comparison to the global painting approach (*RGB-P*), this enhances the overall average detection accuracy by 1.51% and 3.39% for 3D and BEV detection results, respectively.

Ablation Study on Different LiDAR-Based 3D Object Detectors. We employ various LiDAR-based 3D object detectors, including PointPillars [42], PV-RCNN [44], and Voxel-RCNN [43], to showcase the versatility of VFMM3D. The impacts of different 3D detectors on prediction accuracy are illustrated in Tab. II and Tab. I.

V. Conclusion

In this paper, we have introduced VFMM3D, a novel framework for monocular 3D object detection that leverages Vision Foundation Models (VFMs) to unlock the potential of monocular image data for 3D object detection. By integrating the Segment Anything Model (SAM) and Depth Anything Model (DAM), VFMM3D generates pseudo-LiDAR data that enriches the depth features with detailed spatial information without the need for being finetuned on specific datasets. Our proposed approach addresses key challenges in monocular 3D object detection, including computational efficiency and noise reduction. VFMM3D could be seamlessly integrated into diverse LiDAR-based detectors, which enhances its versatility and applicability across detection architectures. Through extensive experiments on the Waymo and KITTI datasets, we have demonstrated the superior performance of VFMM3D over existing methods across various difficulty levels. Our framework consistently outperforms state-of-the-art approaches, showcasing its efficacy in extracting precise 3D object representations from monocular images. The results highlight the potential of VFMM3D as a robust and adaptable solution for real-world deployment in autonomous driving and robotics applications.

In future work, we would further explore the capabilities of VFMM3D by investigating additional datasets in different real-world scenarios, such as indoor and complex weather conditions. Additionally, we will refine our sparsification techniques to improve computational efficiency without compromising detection accuracy.

References

[1] L. Zhao, J. Guo, D. Xu, and L. Sheng, “Transformer3d-det: Improving 3d object detection by vote refinement,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4735–4746, 2021.

[2] M. Song, S. Lim, and W. Kim, “Monocular depth estimation using laplacian pyramid-based depth residuals,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4381–4393, 2021.

[3] S. Chen, Z. Pu, X. Fan, and B. Zou, “Fixing defect of photometric loss for self-supervised monocular depth estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1328–1338, 2022.

[4] H. Sheng, S. Cai, N. Zhao, B. Deng, M.-J. Zhao, and G. H. Lee, “Pdr: Progressive depth regularization for monocular 3d object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7591–7603, 2023.

[5] Y. Cao, Z. Wu, and C. Shen, “Estimating depth from monocular images as classification using deep fully convolutional residual networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3174–3182, 2018.

[6] Y. Chen, L. Tai, K. Sun, and M. Li, “Monopair: Monocular 3d object detection using pairwise spatial relationships,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] J. Ku, A. D. Pon, and S. L. Waslander, “Monocular 3d object detection leveraging accurate proposals and shape reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, “3d bounding box estimation using deep learning and geometry,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[9] Y. Lu, X. Ma, L. Yang, T. Zhang, Y. Liu, Q. Chu, J. Yan, and W. Ouyang, “Geometry uncertainty projection network for monocular 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3111–3121.

[10] X. Shi, Q. Ye, X. Chen, C. Chen, Z. Chen, and T.-K. Kim, “Geometry-based distance decomposition for monocular 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 172–15 181.

[11] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, “Autoshape: Real-time shape-aware monocular 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 641–15 650.

[12] G. Brazil and X. Liu, “M3d-rpn: Monocular 3d region proposal network for object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[13] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3d object detection for autonomous driving,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2147–2156.

[14] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, “Disentangling monocular 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1991–1999.

[15] Y. Zhang, W. Zheng, Z. Zhu, G. Huang, D. Du, J. Zhou, and J. Lu, "Dimension embeddings for monocular 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1589–1598.

[16] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, and W. Ouyang, "Delving into localization errors for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4721–4730.

[17] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8555–8564.

[18] Z. Qin and X. Li, "Monoground: Detecting monocular 3d objects from the ground," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 3793–3802.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[20] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2024.

[21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[22] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[23] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9650–9660.

[24] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 3289–3298.

[25] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning*. PMLR, 2022, pp. 1475–1485.

[26] H. Chen, Y. Huang, W. Tian, Z. Gao, and L. Xiong, "Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 379–10 388.

[27] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "Monodtr: Monocular 3d object detection with depth-aware transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4012–4021.

[28] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao, "Monodetr: Depth-guided transformer for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 9155–9166.

[29] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6851–6860.

[30] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3142–3152.

[31] X. Weng and K. Kitani, "Monocular 3d object detection with pseudo-lidar point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[32] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8445–8453.

[33] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1057–1066.

[34] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 644–660.

[35] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE, 2019, pp. 770–779.

[36] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[37] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1951–1960.

[38] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3d instance segmentation and object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1839–1849.

[39] Q. Wang, J. Chen, J. Deng, and X. Zhang, "3d-centernet: 3d object detection network for point clouds with center estimation priority," *Pattern Recognition*, vol. 115, p. 107884, 2021.

[40] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[41] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[42] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE, 2019, pp. 12 697–12 705.

[43] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI conference on artificial intelligence*, 2021, pp. 1201–1209.

[44] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE, 2020, pp. 10 529–10 538.

[45] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[46] L. Ke, M. Ye, M. Danelljan, Y.-W. Tai, C.-K. Tang, F. Yu *et al.*, "Segment anything in high quality," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[47] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[48] S. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, "Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3047–3054.

[49] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[50] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BJedHRVtPB

[51] H. Zhu, J. Deng, Y. Zhang, J. Ji, Q. Mao, H. Li, and Y. Zhang, "Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion," *IEEE Transactions on Multimedia*, 2022.

[52] H. Wu, C. Wen, S. Shi, X. Li, and C. Wang, "Virtual sparse convolution for multimodal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 653–21 662.

[53] J. Xu, L. Peng, H. Cheng, H. Li, W. Qian, K. Li, W. Wang, and D. Cai, "Mononerd: Nerf-like representations for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 6814–6824.

[54] L. Peng, X. Wu, Z. Yang, H. Liu, and D. Cai, "Did-m3d: Decoupling instance depth for monocular 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 71–88.

[55] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-lidar representation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 311–327.

[56] L. Wang, L. Zhang, Y. Zhu, Z. Zhang, T. He, M. Li, and X. Xue, "Progressive coordinate transforms for monocular 3d object detection,"

Advances in Neural Information Processing Systems, vol. 34, pp. 13 364–13 377, 2021.

[57] A. Kumar, G. Brazil, E. Corona, A. Parchami, and X. Liu, “Deviant: Depth equivariant network for monocular 3d object detection,” in *European Conference on Computer Vision*. Springer, 2022, pp. 664–683.

[58] Q. Lian, P. Li, and X. Chen, “Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1070–1079.

[59] X. He, F. Yang, K. Yang, J. Lin, H. Fu, M. Wang, J. Yuan, and Z. Li, “Ssd-monodetr: Supervised scale-aware deformable transformer for monocular 3d object detection,” *IEEE Transactions on Intelligent Vehicles*, 2023.

[60] L. Wang, L. Du, X. Ye, Y. Fu, G. Guo, X. Xue, J. Feng, and L. Zhang, “Depth-conditioned dynamic message propagation for monocular 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 454–463.

[61] J. Gu, B. Wu, L. Fan, J. Huang, S. Cao, Z. Xiang, and X.-S. Hua, “Homography loss for monocular 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1080–1089.

[62] Z. Li, Z. Qu, Y. Zhou, J. Liu, H. Wang, and L. Jiang, “Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 2791–2800.

[63] Y. Su, Y. Di, G. Zhai, F. Manhardt, J. Rambach, B. Busam, D. Stricker, and F. Tombari, “Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1327–1334, 2023.

[64] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[65] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[66] A. Simonelli, S. R. Bulo, L. Porzi, M. Lopez-Antequera, and P. Kontschieder, “Disentangling monocular 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[67] O. D. Team, “Openpcdet: An open-source toolbox for 3d object detection from point clouds,” https://github.com/open-mmlab/OpenPCDet, 2020.

[68] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.